

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ
по лабораторной работе №5
на тему : «Технический долг. Устранение долга»

Студенты:

Е.С. Рогачевский
А.М. Гнездилов
В.А. Блинников

Руководитель:

Д.А. Жалейко

Минск 2024

Ход работы:

1 Инспекция проекта на наличие технического долга

Проект был проинспектирован на наличие признаков технического долга, и были выявлены следующие проблемы:

1. Медленные / неэффективные средства – некоторые части приложения работают медленно и неэффективно, что сказывается на производительности.

```
@SuppressWarnings("SetTextI18n")
@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    try {
        if (result != null) {
            JSONObject jsonObject = new JSONObject(result);

            // Проверяем, есть ли данные о погоде в JSON-ответе
            if (jsonObject.has( name: "main")) {
                // Получаем данные о погоде
                JSONObject main = jsonObject.getJSONObject( name: "main");
                double humidity = main.getDouble( name: "humidity");

                Calendar calendar = Calendar.getInstance();
                int hour = calendar.get(Calendar.HOUR_OF_DAY);

                // Установка значения в соответствующий ImageView
                LinearLayout linearLayout1 = findViewById(R.id.LinearLayout1);
                LinearLayout linearLayout2 = findViewById(R.id.LinearLayout2);
                LinearLayout linearLayout3 = findViewById(R.id.LinearLayout3);
                LinearLayout linearLayout4 = findViewById(R.id.LinearLayout4);
                LinearLayout linearLayout5 = findViewById(R.id.LinearLayout5);
                LinearLayout linearLayout6 = findViewById(R.id.LinearLayout6);
                LinearLayout linearLayout7 = findViewById(R.id.LinearLayout7);
                LinearLayout linearLayout8 = findViewById(R.id.LinearLayout8);
                LinearLayout linearLayout9 = findViewById(R.id.LinearLayout9);
                LinearLayout linearLayout10 = findViewById(R.id.LinearLayout10);
                LinearLayout linearLayout11 = findViewById(R.id.LinearLayout11);
                LinearLayout linearLayout12 = findViewById(R.id.LinearLayout12);

                ImageView humidityImageView1 = linearLayout1.findViewById(R.id.humidityImageView);
                ImageView humidityImageView2 = linearLayout2.findViewById(R.id.humidityImageView2);
                ImageView humidityImageView3 = linearLayout3.findViewById(R.id.humidityImageView3);
                ImageView humidityImageView4 = linearLayout4.findViewById(R.id.humidityImageView4);
                ImageView humidityImageView5 = linearLayout5.findViewById(R.id.humidityImageView5);
                ImageView humidityImageView6 = linearLayout6.findViewById(R.id.humidityImageView6);
                ImageView humidityImageView7 = linearLayout7.findViewById(R.id.humidityImageView7);
                ImageView humidityImageView8 = linearLayout8.findViewById(R.id.humidityImageView8);
                ImageView humidityImageView9 = linearLayout9.findViewById(R.id.humidityImageView9);
                ImageView humidityImageView10 = linearLayout10.findViewById(R.id.humidityImageView10);
                ImageView humidityImageView11 = linearLayout11.findViewById(R.id.humidityImageView11);
                ImageView humidityImageView12 = linearLayout12.findViewById(R.id.humidityImageView12);
```

```
if (hour >= 0 && hour < 2) {
    if (humidity <= 20) {
        humidityImageView1.setImageResource(R.drawable.solnechno);
    } else if (humidity <= 40) {
        humidityImageView1.setImageResource(R.drawable.zatyanyto);
    } else if (humidity <= 60) {
        humidityImageView1.setImageResource(R.drawable.doshdlivo);
    } else {
        humidityImageView1.setImageResource(R.drawable.liveni);
    }
} else if (hour >= 2 && hour < 4) {
    if (humidity <= 20) {
        humidityImageView2.setImageResource(R.drawable.solnechno);
    } else if (humidity <= 40) {
        humidityImageView2.setImageResource(R.drawable.zatyanyto);
    } else if (humidity <= 60) {
        humidityImageView2.setImageResource(R.drawable.doshdlivo);
    } else {
        humidityImageView2.setImageResource(R.drawable.liveni);
    }
} else if (hour >= 4 && hour < 6) {
    if (humidity <= 20) {
        humidityImageView3.setImageResource(R.drawable.solnechno);
    } else if (humidity <= 40) {
        humidityImageView3.setImageResource(R.drawable.zatyanyto);
    } else if (humidity <= 60) {
        humidityImageView3.setImageResource(R.drawable.doshdlivo);
    } else {
        humidityImageView3.setImageResource(R.drawable.liveni);
    }
} else if (hour >= 6 && hour < 8) {
    if (humidity <= 20) {
        humidityImageView4.setImageResource(R.drawable.solnechno);
    } else if (humidity <= 40) {
        humidityImageView4.setImageResource(R.drawable.zatyanyto);
    } else if (humidity <= 60) {
        humidityImageView4.setImageResource(R.drawable.doshdlivo);
    } else {
        humidityImageView4.setImageResource(R.drawable.liveni);
    }
} else if (hour >= 8 && hour < 10) {
    if (humidity <= 20) {
        humidityImageView5.setImageResource(R.drawable.solnechno);
    } else if (humidity <= 40) {
        humidityImageView5.setImageResource(R.drawable.zatyanyto);
    } else if (humidity <= 60) {
        humidityImageView5.setImageResource(R.drawable.doshdlivo);
    } else {
        humidityImageView5.setImageResource(R.drawable.liveni);
    }
} else if (hour >= 10 && hour < 12) {
    if (humidity <= 20) {
        humidityImageView6.setImageResource(R.drawable.solnechno);
    } else if (humidity <= 40) {
        humidityImageView6.setImageResource(R.drawable.zatyanyto);
    } else if (humidity <= 60) {
        humidityImageView6.setImageResource(R.drawable.doshdlivo);
    } else {
        humidityImageView6.setImageResource(R.drawable.liveni);
    }
}
```

```

    } else if (hour >= 20 && hour < 22) {
        if (humidity <= 20) {
            humidityImageView11.setImageResource(R.drawable.solnechno);
        } else if (humidity <= 40) {
            humidityImageView11.setImageResource(R.drawable.zatyanyto);
        } else if (humidity <= 60) {
            humidityImageView11.setImageResource(R.drawable.doshdlivo);
        } else {
            humidityImageView11.setImageResource(R.drawable.liveni);
        }
    } else if (hour >= 22 && hour < 24) {
        if (humidity <= 20) {
            humidityImageView12.setImageResource(R.drawable.solnechno);
        } else if (humidity <= 40) {
            humidityImageView12.setImageResource(R.drawable.zatyanyto);
        } else if (humidity <= 60) {
            humidityImageView12.setImageResource(R.drawable.doshdlivo);
        } else {
            humidityImageView12.setImageResource(R.drawable.liveni);
        }
    }
}
}
} catch (JSONException e) {
    e.printStackTrace();
}
}
}

```

Рисунок 1.1. Медленные / неэффективные. До исправления

```

@SuppressLint("SetTextI18n")
@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);
    try {
        if (result != null) {
            JSONObject jsonObject = new JSONObject(result);

            // Проверяем, есть ли данные о погоде в JSON-ответе
            if (jsonObject.has( name: "main")) {
                // Получаем данные о погоде
                JSONObject main = jsonObject.getJSONObject( name: "main");
                double humidity = main.getDouble( name: "humidity");
                int roundedClouds = (int) Math.round(humidity);

                Calendar calendar = Calendar.getInstance();
                int hour = calendar.get(Calendar.HOUR_OF_DAY);

                // Установка значения в соответствующий ImageView
                LinearLayout linearLayout;
                ImageView humidityImageView;

                if (hour >= 0 && hour < 2) {
                    linearLayout = findViewById(R.id.LinearLayout1);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView);
                } else if (hour >= 2 && hour < 4) {
                    linearLayout = findViewById(R.id.LinearLayout2);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView2);
                } else if (hour >= 4 && hour < 6) {
                    linearLayout = findViewById(R.id.LinearLayout3);

```

```

        humidityImageView = linearLayout.findViewById(R.id.humidityImageView3);
    } else if (hour >= 6 && hour < 8) {
        linearLayout = findViewById(R.id.LinearLayout4);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView4);
    } else if (hour >= 8 && hour < 10) {
        linearLayout = findViewById(R.id.LinearLayout5);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView5);
    } else if (hour >= 10 && hour < 12) {
        linearLayout = findViewById(R.id.LinearLayout6);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView6);
    } else if (hour >= 12 && hour < 14) {
        linearLayout = findViewById(R.id.LinearLayout7);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView7);
    } else if (hour >= 14 && hour < 16) {
        linearLayout = findViewById(R.id.LinearLayout8);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView8);
    } else if (hour >= 16 && hour < 18) {
        linearLayout = findViewById(R.id.LinearLayout9);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView9);
    } else if (hour >= 18 && hour < 20) {
        linearLayout = findViewById(R.id.LinearLayout10);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView10);
    } else if (hour >= 20 && hour < 22) {
        linearLayout = findViewById(R.id.LinearLayout11);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView11);
    } else {
        linearLayout = findViewById(R.id.LinearLayout12);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView12);
    }
}

        if (roundedClouds <= 10) {
            humidityImageView.setImageResource(R.drawable.solnechno);
        } else if (roundedClouds <= 20) {
            humidityImageView.setImageResource(R.drawable.zatyanyto);
        } else if (roundedClouds <= 30) {
            humidityImageView.setImageResource(R.drawable.doshdlivo);
        } else {
            humidityImageView.setImageResource(R.drawable.liveni);
        }
    }
} catch (JSONException e) {
    e.printStackTrace();
}
}

```

Рисунок 1.2. Медленные / неэффективные. После исправления

2. Непонятный / нечитабельный код – в некоторых местах код написан небрежно, с использованием неочевидных имен переменных и отсутствием комментариев.

```

connection = (URLConnection) url.openConnection();

// connection.connect();
//
//      InputStream stream = connection.getInputStream();
//      reader = new BufferedReader(new InputStreamReader(stream));
//
//
//
//      StringBuilder buffer = new StringBuilder();
//      String line;
//      connection.connect();

InputStream stream = connection.getInputStream();
reader = new BufferedReader(new InputStreamReader(stream));

StringBuilder buffer = new StringBuilder();
String line;

while ((line = reader.readLine()) != null) {
    buffer.append(line).append("\n");
}

String data = buffer.toString();
Log.d( tag: "JSON_DATA", data); // Добавленный код

```

Рисунок 1.3. Непонятный / нечитабельный код. До исправления

```

@Override
protected String doInBackground(String... strings) {
    HttpURLConnection connection = null;
    BufferedReader reader = null;
    try {
        URL url = new URL(strings[0]);
        connection = (URLConnection) url.openConnection();
        connection.connect();

        InputStream stream = connection.getInputStream();
        reader = new BufferedReader(new InputStreamReader(stream));

        StringBuilder buffer = new StringBuilder();
        String line;

        while ((line = reader.readLine()) != null) {
            buffer.append(line).append("\n");
        }

        String data = buffer.toString();
        Log.d( tag: "JSON_DATA", data); // Добавленный код

        return data;

    } catch (IOException e) {
        e.printStackTrace();
        return null;
    } finally {

```

Рисунок 1.4. Непонятный / нечитабельный код. После исправления

3. Незакоммиченный код / долгоживущие ветки — часть кода не закоммичена, а некоторые ветки существуют слишком долго, что приводит к рассинхронизации и затрудняет совместную работу.

```

/*protected void onPostExecute(String result) {
    super.onPostExecute(result);
    try {
        if (result != null) {
            JSONObject jsonObject = new JSONObject(result);

            // Проверяем, есть ли данные о погоде в JSON-ответе
            if (jsonObject.has("main")) {
                // Получаем данные о погоде
                JSONObject main = jsonObject.getJSONObject("main");
                double humidity = main.getDouble("humidity");
                int roundedClouds = (int) Math.round(humidity);

                Calendar calendar = Calendar.getInstance();
                int hour = calendar.get(Calendar.HOUR_OF_DAY);

                // Установка значения в соответствующий ImageView
                LinearLayout linearLayout;
                ImageView humidityImageView;

                if (hour >= 0 && hour < 2) {
                    linearLayout = findViewById(R.id.LinearLayout1);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView);
                } else if (hour >= 2 && hour < 4) {
                    linearLayout = findViewById(R.id.LinearLayout2);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView2);
                } else if (hour >= 4 && hour < 6) {
                    linearLayout = findViewById(R.id.LinearLayout3);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView3);
                } else if (hour >= 6 && hour < 8) {
                    linearLayout = findViewById(R.id.LinearLayout4);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView4);
                } else if (hour >= 8 && hour < 10) {
                    linearLayout = findViewById(R.id.LinearLayout5);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView5);
                } else if (hour >= 10 && hour < 12) {
                    linearLayout = findViewById(R.id.LinearLayout6);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView6);
                } else if (hour >= 12 && hour < 14) {
                    linearLayout = findViewById(R.id.LinearLayout7);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView7);
                } else if (hour >= 14 && hour < 16) {
                    linearLayout = findViewById(R.id.LinearLayout8);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView8);
                } else if (hour >= 16 && hour < 18) {
                    linearLayout = findViewById(R.id.LinearLayout9);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView9);
                } else if (hour >= 18 && hour < 20) {
                    linearLayout = findViewById(R.id.LinearLayout10);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView10);
                } else if (hour >= 20 && hour < 22) {
                    linearLayout = findViewById(R.id.LinearLayout11);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView11);
                } else {
                    linearLayout = findViewById(R.id.LinearLayout12);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView12);
                }
            }
        }
    }
}

```

```

        if (roundedClouds <= 10) {
            humidityImageView.setImageResource(R.drawable.solnechno);
        } else if (roundedClouds <= 20) {
            humidityImageView.setImageResource(R.drawable.zatyanyto);
        } else if (roundedClouds <= 30) {
            humidityImageView.setImageResource(R.drawable.doshdlivo);
        } else {
            humidityImageView.setImageResource(R.drawable.liveni);
        }
    }
}
} catch (JSONException e) {
    e.printStackTrace();
}
}
} \*

```

Рисунок 1.5. Незаконмиченный код / долгоживущие ветки. До исправления

```

protected void onPostExecute(String result) {
    super.onPostExecute(result);
    try {
        if (result != null) {
            JSONObject jsonObject = new JSONObject(result);

            // Проверяем, есть ли данные о погоде в JSON-ответе
            if (jsonObject.has( name: "main")) {
                // Получаем данные о погоде
                JSONObject main = jsonObject.getJSONObject( name: "main");
                double humidity = main.getDouble( name: "humidity");
                int roundedClouds = (int) Math.round(humidity);

                Calendar calendar = Calendar.getInstance();
                int hour = calendar.get(Calendar.HOUR_OF_DAY);

                // Установка значения в соответствующий ImageView
                LinearLayout linearLayout;
                ImageView humidityImageView;

                if (hour >= 0 && hour < 2) {
                    linearLayout = findViewById(R.id.LinearLayout1);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView);
                } else if (hour >= 2 && hour < 4) {
                    linearLayout = findViewById(R.id.LinearLayout2);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView2);
                } else if (hour >= 4 && hour < 6) {
                    linearLayout = findViewById(R.id.LinearLayout3);
                    humidityImageView = linearLayout.findViewById(R.id.humidityImageView3);
                } else if (hour >= 6 && hour < 8) {
                    linearLayout = findViewById(R.id.LinearLayout4);

```



```

        humidityImageView = linearLayout.findViewById(R.id.humidityImageView4);
    } else if (hour >= 8 && hour < 10) {
        linearLayout = findViewById(R.id.LinearLayout5);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView5);
    } else if (hour >= 10 && hour < 12) {
        linearLayout = findViewById(R.id.LinearLayout6);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView6);
    } else if (hour >= 12 && hour < 14) {
        linearLayout = findViewById(R.id.LinearLayout7);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView7);
    } else if (hour >= 14 && hour < 16) {
        linearLayout = findViewById(R.id.LinearLayout8);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView8);
    } else if (hour >= 16 && hour < 18) {
        linearLayout = findViewById(R.id.LinearLayout9);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView9);
    } else if (hour >= 18 && hour < 20) {
        linearLayout = findViewById(R.id.LinearLayout10);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView10);
    } else if (hour >= 20 && hour < 22) {
        linearLayout = findViewById(R.id.LinearLayout11);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView11);
    } else {
        linearLayout = findViewById(R.id.LinearLayout12);
        humidityImageView = linearLayout.findViewById(R.id.humidityImageView12);
    }
}

        if (roundedClouds <= 10) {
            humidityImageView.setImageResource(R.drawable.solnechno);
        } else if (roundedClouds <= 20) {
            humidityImageView.setImageResource(R.drawable.zatyanyto);
        } else if (roundedClouds <= 30) {
            humidityImageView.setImageResource(R.drawable.doshdlivo);
        } else {
            humidityImageView.setImageResource(R.drawable.liveni);
        }
    }
}

} catch (JSONException e) {
    e.printStackTrace();
}
}

```

Рисунок 1.6. Незакоммиченный код / долгоживущие ветки. После исправления

Графическое изображение технического долга, который наблюдался в процессе разработки проекта:

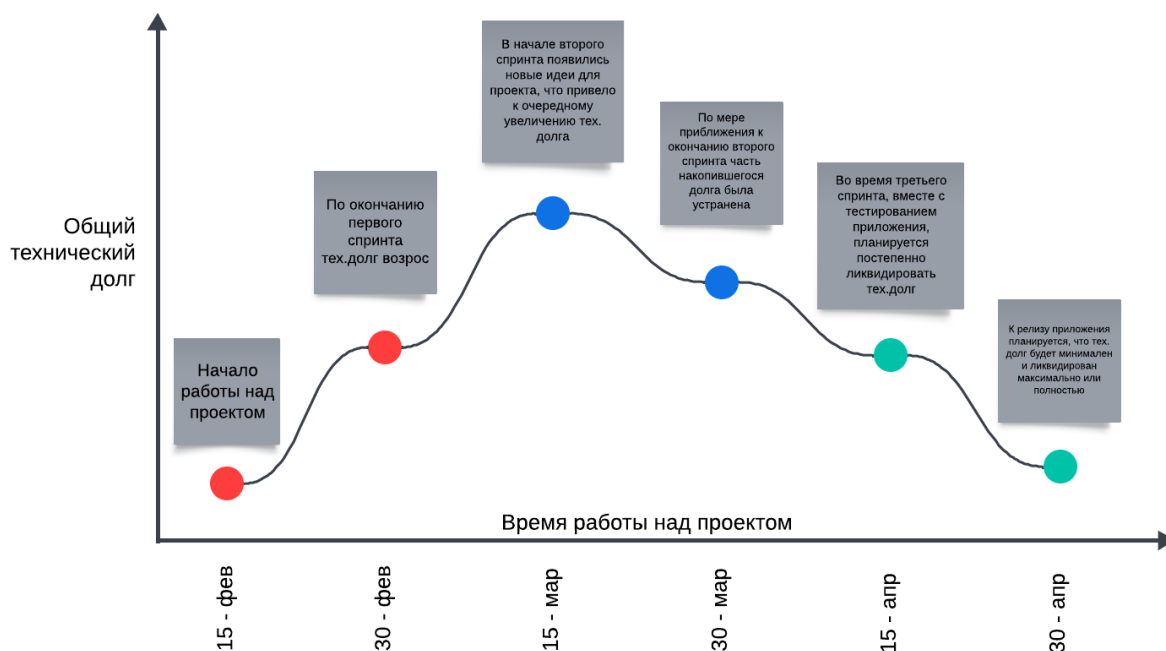


Рисунок 1.7. Графическое изображение технического долга

2 План мероприятий по устранению технического долга

| Мероприятие | Описание | Приоритет |
|-----------------------------------|--|-----------|
| Рефакторинг кода | Улучшить читаемость кода, используя более понятные имена переменных, добавляя комментарии и реорганизуя код. | Высокий |
| Устранение дублирования кода | Выделить повторяющийся код в отдельные функции или модули, чтобы уменьшить избыточность и упростить поддержку. | Высокий |
| Внедрение автоматизации | Создать автоматические тесты, настроить сборку и развертывание, чтобы уменьшить ручной труд и количество ошибок. | Высокий |
| Улучшение архитектуры | Переработать архитектуру приложения, чтобы сделать ее более понятной и эффективной. | Средний |
| Оптимизация производительности | Проанализировать и оптимизировать код, чтобы повысить производительность приложения. | Средний |
| Внедрение системы контроля версий | Использовать систему контроля версий для отслеживания изменений кода и облегчения совместной работы. | Средний |

| | | |
|--------------------------------------|---|---------|
| Создание и актуализация документации | Создать и поддерживать актуальную техническую документацию, чтобы облегчить понимание и поддержку кода. | Средний |
| Создание тестовой среды | Создать выделенную тестовую среду для тестирования и отладки приложения. | Средний |
| Внедрение непрерывной интеграции | Настроить непрерывную интеграцию, чтобы новый код интегрировался и тестировался автоматически. | Средний |

3 Оценка плана мероприятий

План мероприятий по устранению технического долга оценивается как **необходимый**.

Устранение технического долга улучшит качество кода, повысит производительность приложения, упростит его поддержку и разработку новых функций.

4 Сравнение объема долга и недоимплементированных фич

Объем технического долга в проекте **значителен**.

Некоторые недоимплементированные фичи могут быть отложены, но устранение технического долга необходимо для дальнейшей разработки и поддержки приложения.

5 Выводы и коррективы в план разработки

Технический долг в проекте **существенен и требует внимания**.

Устранение технического долга должно быть приоритетом в плане разработки, так как это улучшит качество кода, повысит производительность приложения, упростит его поддержку и разработку новых функций.

Рекомендуется:

- Внести коррективы в план разработки, чтобы выделить время на устранение технического долга.
- Начать с задач с высоким приоритетом, таких как рефакторинг кода, устранение дублирования кода и внедрение автоматизации.
- Постепенно устранять остальные проблемы.

Важно найти баланс между устранением технического долга и разработкой новых функций.

Полное игнорирование технического долга может привести к серьезным проблемам в будущем.